Heart Attack Risk Prediction

Decision Trees, Random Forest & AdaBoost



**Submitted By:**

* Coral Borafker – 322262221
* Tal Gorodetzky – 211898739
* Rachel Harow – 332398809

**Instructor:** Dr. [Yehuda Hassin](https://www.bing.com/ck/a?!&&p=6f962d403f23b5e1ef6fe6c7844ef39fbd29ba3e807966b7f9f49daf8783192fJmltdHM9MTc1Mzc0NzIwMA&ptn=3&ver=2&hsh=4&fclid=0c6c6e07-8edc-6653-3624-7eff8f4c677f&psq=dr+yehuda+hassin+azrieli+college&u=a1aHR0cHM6Ly9qY2VuZXcubWlwby5jby5pbC9kci15ZWh1ZGEtaGFzc2luLw&ntb=1)

**Institution:** Azrieli - College of Engineering Jerusalem

**Course Code:** 10123

**Submission Date:** July 29, 2025

# 1. Introduction

This project focuses on the application of **machine learning algorithms,** specifically predicting **heart attack risk** based on various medical, demographic, and lifestyle factors. The aim was to **implement three fundamental supervised learning algorithms from scratch**—Decision Trees, Random Forest, and AdaBoost—and compare their performance with scikit-learn equivalents.

The primary objective was not only to achieve good predictive accuracy but also to **understand the inner workings of these algorithms**, their strengths, and their trade-offs in terms of accuracy, interpretability, and robustness.

# 2. Dataset Description

The dataset, **"heart\_attack\_prediction\_dataset.csv"**, was sourced from **Kaggle**.  
It contains **8,763 samples and 23 features**, including a mix of numerical and categorical variables. The target variable is:

* **Heart Attack Risk:** Binary label (0 = Low Risk, 1 = High Risk)

**Key Feature Categories**

* **Demographic:** Age, Sex, Income
* **Clinical:** Cholesterol, Blood Pressure, Heart Rate, BMI, Triglycerides
* **Lifestyle:** Exercise hours, Physical activity days, Sedentary hours, Diet
* **Medical history:** Diabetes, Family history, Medication use, Previous heart problems
* **Behavioral:** Smoking, Alcohol consumption
* **Psychological:** Stress level

# 3. Methodology

### 3.1 Data Preprocessing

* Converted categorical features to numeric form.
* Ensured all values were within realistic ranges (e.g., age between 0–120).
* Checked for outliers and normalized data where necessary.
* Split the dataset into **70% training** and **30% testing** subsets using a fixed random\_state=42.

### 3.2 Algorithm Implementations

#### 3.2.1 Decision Tree (Custom Implementation)

* **Splitting Criterion:** Gini impurity
* **Stopping Rules:** Max depth and minimum samples per leaf
* **Prediction:** Recursive traversal from root to leaf nodes

#### 3.2.2 Random Forest (Custom Implementation)

* **Bagging:** Bootstrap samples used for each tree.
* **Feature Subsampling:** Random subset of features for each split.
* **Voting:** Predictions aggregated via majority voting.
* **Benefit:** Reduced variance and improved generalization compared to a single tree.

#### 3.2.3 AdaBoost (Custom Implementation)

* **Weak Learners:** Depth-2 decision stumps.
* **Weighted Iterations:** Focus on misclassified samples by increasing their weights.
* **Final Prediction:** Weighted vote of all weak learners.

# 4. Experimental Results

We trained all three algorithms on the same dataset and compared **manual implementations** against **scikit-learn's versions**.

### 4.1 Accuracy Results

| **Algorithm** | **Manual Accuracy** | **Sklearn Accuracy** |
| --- | --- | --- |
| 🧠 Decision Tree | **0.6417** | 0.6405 |
| 🌲 Random Forest | **0.6428** | 0.6383 |
| 🔧 AdaBoost | **0.6432** | 0.6413 |

### 4.2 Analysis

* **Decision Tree:** High interpretability but prone to overfitting.
* **Random Forest:** Improved generalization due to ensemble averaging, best for stability.
* **AdaBoost:** Slightly better accuracy by focusing on misclassified samples, but more sensitive to noise.

# 5. Discussion and Conclusions

## 5.1 Key Findings

* **Random Forest and AdaBoost outperformed a single Decision Tree**, showing the advantage of ensemble methods.
* **Manual implementations closely matched scikit-learn models**, validating the correctness of our code.
* Tree-based methods provide good interpretability, making them suitable for risk prediction tasks where decision transparency is important.

### 5.2 Insights from Manual Implementation

* Understanding **splitting criteria, information gain, and ensemble aggregation** deepened our comprehension of supervised learning.
* Implementing algorithms from scratch revealed practical considerations like stopping conditions, and handling of categorical features.

# 6. Conclusion

This project successfully demonstrated the design and implementation of three classic machine learning algorithms for binary classification.  
While **Random Forest and AdaBoost achieved slightly better accuracy**, all models provided interpretable predictions and competitive results.

The manual implementation process strengthened understanding of **core machine learning principles**, particularly decision boundaries, bias-variance trade-offs, and the power of ensemble methods.